WHAT IS CLAIMED IS:

1. A method comprising:

adding a first plurality of data elements to a second plurality of data elements generating a plurality of intermediate results;

adding two of the plurality of intermediate results and repeating with different combinations of the plurality of intermediate results generating a plurality of sum results; and

discarding the two least significant bits of each sum result of the plurality of sum results.

2. The method as recited in Claim 1, further comprising:

performing a carry in of a value of one when performing the adding the first plurality of data elements to the second plurality of data elements.

3. The method as recited in Claim 1, further comprising:

performing a carry in of a rounding term when performing the adding the two of the plurality of intermediate results and the repeating.

4. The method as recited in Claim 3, wherein the rounding term is a variable capable of having a value of one and, at a different time, a value of zero.

5. The method as recited in Claim 1, further comprising:

performing a carry in of a value of one when adding the first plurality of data elements to the second plurality of data elements; and

performing a carry in of a rounding term when adding the two of the plurality of intermediate results and when repeating.

6. The method as recited in Claim 1, wherein the first plurality of data elements and the second plurality of data elements each comprise eight eight-bit unsigned data elements.

7. The method as recited in Claim 1, wherein the first plurality of data elements and the second plurality of data elements each comprise eight sixteen-bit data elements.

8. The method as recited in Claim 1, wherein the method comprises executing a Single-Instruction/Multiple-Data (SIMD) instruction.

9. The method as recited in Claim 1, wherein the method is performed utilizing Single-Instruction/Multiple-Data (SIMD) circuitry.

10. A method comprising:

adding an i_{th} data element of a first source to an i_{th} data element of a second source creating an i_{th} intermediate result for i = 1 to N, wherein N is an integer greater than 1;

adding a j_{th} intermediate result to a $(j+1)_{th}$ intermediate result creating a j_{th} sum result for j = 1 to (N-1); and

discarding two least significant bits of each j_{th} sum result.

11. The method as recited in Claim 10, further comprising:

performing a carry in of a value of one when adding the i_{th} data element of the first source to the i_{th} data element of the second source; and

performing a carry in of a rounding term when adding the j_{th} intermediate result to the $(j+1)_{th}$ intermediate result.

12. The method as recited in Claim 11, wherein the rounding term is selected from a group consisting of a value of one and a value of zero.

13. The method as recited in Claim 10, wherein N = 8.

14. The method as recited in Claim 10, wherein the method is performed during execution of a Single-Instruction/Multiple-Data (SIMD) instruction.

15. An apparatus comprising:

a plurality of first adders, each first adder of the plurality of first adders operative to add two operands of a plurality of operands into one of a plurality of intermediate results;

a plurality of second adders, each second adder of the plurality of second adders operative to add two intermediate results of the plurality of intermediate results into one of a plurality of sum results; and

discard circuitry operative to discard the two least significant bits of each sum result of the plurality of sum results.

16. The apparatus as recited in Claim 15, wherein the plurality of first adders comprises eight first adders and the plurality of second adders comprises seven second adders.

17. The apparatus as recited in Claim 15, wherein the discard circuitry comprises a plurality of shift registers.

18. The apparatus as recited in Claim 15, wherein each of the first adders are operative to add two eight-bit input operands producing a nine-bit intermediate operand and each of the second adders are operative to add two nine-bit intermediate operands producing a ten-bit output operand.

19. The apparatus as recited in Claim 15, wherein each of the first adders are operative to add two sixteen-bit input operands producing a seventeen-bit intermediate operand and each of the second adders are operative to add two seventeen-bit intermediate operands producing an eighteen-bit operand.

20. The apparatus as recited in Claim 15, wherein routing of the plurality of operands and the plurality of intermediate results to the plurality of first adders and the plurality of second adders is selected according to microcode identified by a Single-Instruction/Multiple-Data (SIMD) instruction.

21. The apparatus as recited in Claim 15, wherein routing of the plurality of operands and the plurality of intermediate results to the plurality of first adders and the

plurality of second adders is selected according to decode logic and a Single-Instruction/Multiple-Data (SIMD) instruction.

22. The apparatus as recited in Claim 15, wherein the plurality of first adders, the plurality of second adders, and the discard circuitry form a Single-Instruction/Multiple-Data (SIMD) instruction execution circuit.

23. An apparatus comprising:

a plurality of first adders operative to add an i_{th} data element of a first source to an i_{th} data element of a second source generating an i_{th} intermediate result for i = 1 to N, wherein N is an integer greater than 1;

a plurality of second adders operative to add a j_{th} intermediate result to a $(j+1)_{th}$ intermediate result generating a j_{th} sum result for j = 1 to (N-1); and

circuitry operative to discard two least significant bits of each j_{th} sum result.

24. The apparatus as recited in Claim 23, wherein the circuitry comprises a plurality of shift registers.

25. The apparatus as recited in Claim 23, wherein routing of the plurality of operands and the plurality of intermediate results to the plurality of first adders and the plurality of second adders is selected according to microcode identified by a Single-Instruction/Multiple-Data (SIMD)instruction.

26. A method comprising:

decoding an instruction identifying an averaging operation;

executing the instruction on a first source and a second source, wherein the first source comprises a first plurality of data elements and the second source comprises a second plurality of data elements; and

storing a result, wherein the result comprises a third plurality of data elements;

wherein the executing the instruction comprises:

adding successive ones of the first plurality of data elements to successive ones of the second plurality of data elements generating a plurality of intermediate results;

adding two of the plurality of intermediate results and repeating with different combinations of the plurality of intermediate results generating a plurality of sum; and

discarding the two least significant bits of each sum result of the plurality of sum results generating the result.

27. The method as recited in Claim 26, wherein the executing the instruction further comprises:

performing a carry in of a value of one when adding the successive ones of the first plurality of data elements to the successive ones of the second plurality of data elements; and

performing a carry in of a rounding term when adding the two of the plurality of intermediate results and when repeating.

28. The method as recited in Claim 27, wherein the rounding term is selected from a group consisting of a value of one and a value of zero.

29. An apparatus comprising:

a coprocessor interface unit to identify an instruction for an averaging operation, a first source having a first plurality of data elements and a second source having a second plurality of data elements;

an execution unit to perform the averaging operation on the first plurality of data elements and the second plurality of data elements; and

a register to store a result having a third plurality of data elements;

wherein the execution unit is operative to:

add successive ones of the first plurality of data elements to successive ones of the second plurality of data elements generating a plurality of intermediate results;

add two of the plurality of intermediate results and repeating with different combinations of the plurality of intermediate results generating a plurality of sum results; and

discard the two least significant bits of each sum result of the plurality of sum results forming the result.

30. The apparatus as recited in Claim 29, wherein the execution unit is further operative to:

perform a carry in of a value of one when adding the successive ones of the first plurality of data elements to the successive ones of the second plurality of data elements; and

perform a carry in of a rounding term when adding the two of the plurality of intermediate results and when repeating.

31. The apparatus as recited in Claim 30, wherein the rounding term is selected from a group consisting of a value of one and a value of zero.

32. A data processing system comprising:

an addressable memory to store an instruction for an averaging operation;

a processing core coupled to the addressable memory, the processor core comprising:

an execution core to access the instruction;

a first source register to store a first plurality of data elements;

a second source register to store a second plurality of data elements; and

a destination register to store a plurality of results of the averaging operation;

a wireless interface to receive a digital signal comprising a third plurality of data elements; and

an I/O system to provide the first and second plurality of data elements to the first and second source registers from the third plurality of data elements;

wherein the execution core is operative to:

add successive ones of the first plurality of data elements to successive ones of the second plurality of data elements generating a plurality of intermediate results;

add two of the plurality of intermediate results and repeating with different combinations of the plurality of intermediate results generating a plurality of sum results; and

discard the two least significant bits of each sum result of the plurality of sum results generating the plurality of results.

33. The data processing system as recited in Claim 32, wherein the execution core is further operative to:

perform a carry in of a value of one when adding the successive ones of the first plurality of data elements to the successive ones of the second plurality of data elements; and

perform a carry in of a rounding term when adding the two of the plurality of intermediate results and when repeating.

34. The data processing system as recited in Claim 33, wherein the rounding term is a variable capable of having a value of one and, at a different time, a value of zero.

35. An article comprising a machine-readable medium that includes machine readable instructions, the instructions operative to cause a machine to:

add a first plurality of data elements to a second plurality of data elements generating a plurality of intermediate results;

add two of the plurality of intermediate results and repeating with different combinations of the plurality of intermediate results generating a plurality of sum results; and

discard the two least significant bits of each sum result of the plurality of sum results generating a result.

36. The article as recited in Claim 35, the instructions further operative to:

perform a carry in of a value of one when adding the first plurality of data elements to the second plurality of data elements; and

perform a carry in of a rounding term when adding the two of the plurality of intermediate results and when repeating.